UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/082,776	02/25/2002	Victor A. Bennett	BENNETT 6-5	4410

47396 7590 04/03/2006

HITT GAINES, PC
AGERE SYSTEMS INC.
PO BOX 832570
RICHARDSON, TX 75083

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 04/03/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/082,776	BENNETT ET AL.	
	Examiner	Art Unit	
	Aimee J. Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 January 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-21 have been considered. Claims 1, 5, 8, 12, 15, and 19 have been amended as per Applicant's request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received 06 January 2006.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-3, 6-10, 13-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, U.S. Patent Number 5,933,627 (herein referred to as Parady) in view of Microware™'s "OS-9: Real-Time Operating System – The Complete Software Solution for Your Embedded Application" ©1999 (herein referred to as OS-9).

5. Referring to claim 1, Parady has taught a context switching system for a multi-thread execution pipeline loop having a pipeline latency, comprising:

a. A context switch requesting subsystem configured to:

i. Detect a device request from thread executing within said multi-thread execution pipeline loop for access to a device having a fulfillment latency exceeding said pipeline latency (Parady Abstract; column 1, lines 29-35

and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), and

ii. Generate a context Switch request for said thread (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3); and

b. A context controller subsystem configured to receive said context switch request (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3).

6. Parady has not explicitly taught

a. A miss fulfillment first-in-first-out buffer (FIFO); and

b. Based thereon, store said thread in said miss fulfillment FIFO for sequencing there through to prevent said thread from executing until said device request is fulfilled.

7. However, Parady has taught using a round robin scheduling method (Parady column 4, lines 9-11). OS-9 has taught round robin scheduling (OS-9 page 5) with a

a. A miss fulfillment first-in-first-out buffer (FIFO) (OS-9 page 5); and

b. Based thereon, store said thread in said miss fulfillment FIFO for sequencing there through to prevent said thread from executing until said device request is fulfilled (OS-9 page 5).

8. In regards to OS-9, the event queue stores processes that are blocked, waiting for some event to complete, like an I/O access. This is similar to a miss fulfillment FIFO, which, by its name and the language in the claim, stores processes that wait for some other event, like an I/O

access, to complete. A person of ordinary skill in the art at the time the invention was made would have recognized that a round-robin scheduling method is a fairly simple method that guarantees that all processes will be executed while ensuring that priority between processes is maintained, thereby preventing starvation of a process while giving consideration to more important processes. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the round robin method of OS-9 in the device to ensure all processes are executed while maintaining priority and simplicity.

9. Referring to claim 8, Parady has taught for use with a multi-thread execution pipeline loop having a pipeline latency, a method of operating a context switching system, comprising:

a. Detecting a device request from a thread executing within said multi-thread execution pipeline loop for access to a device having a fulfillment latency exceeding said pipeline latency (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3);

b. Generating a context switch request for said thread when said thread issues said device request (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3); and

c. Receiving said context switch request (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3).

10. Parady has not explicitly taught storing said thread based thereon in a miss fulfillment first-in-first-out buffer (FIFO) for sequencing there through until said device request is fulfilled. However, Parady has taught using a round robin scheduling method (Parady column 4, lines 9-11). OS-9 has taught round robin scheduling (OS-9 page 5) storing said thread based thereon in a miss fulfillment first-in-first-out buffer (FIFO) for sequencing there through until said device request is fulfilled (OS-9 page 5). In regards to OS-9, the event queue stores processes that are blocked, waiting for some event to complete, like an I/O access. This is similar to a miss fulfillment FIFO, which, by its name and the language in the claim, stores processes that wait for some other event, like an I/O access, to complete. A person of ordinary skill in the art at the time the invention was made would have recognized that a round-robin scheduling method is a fairly simple method that guarantees that all processes will be executed while ensuring that priority between processes is maintained, thereby preventing starvation of a process while giving consideration to more important processes. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the round robin method of OS-9 in the device to ensure all processes are executed while maintaining priority and simplicity.

11. Referring to claims 2 and 9, Parady in view of OS-9 has taught wherein said context controller subsystem is further configured to allow a new thread to enter said multi-thread execution pipeline loop after storing said thread in said misfulfillment FIFO (OS-9 page 5). In regards to OS-9, after waiting for the event, e.g. I/O access, to complete, the process is moved to the active queue, which holds processes active and ready to be executed (OS-9 page 5). From the active queue, the highest priority process runs each tick (OS-9 page 5).

12. Referring to claims 3 and 10, Parady in view of OS-9 has taught wherein said context controller subsystem is further configured to allow other threads within said multi-thread execution pipeline loop to continue to execute while said thread is waiting for said device request to be fulfilled (Parady column 2, lines 28-34; column 3, line 57 to column 4, line 18; column 4, line 42-62; and Figure 3).

13. Referring to claims 4 and 11, Parady in view of OS-9 has taught wherein said context controller subsystem is further configured to:

a. Store said thread in said miss fulfillment FIFO upon reaching an end position of said multi-thread execution pipeline loop (OS-9 page 5),

b. Sequence said thread through said miss fulfillment FIFO (OS-9 page 5), and

c. Reinsert said thread into said multi-thread execution pipeline loop at a beginning position (OS-9 page 5).

14. Referring to claims 5 and 12, Parady in view of OS-9 has taught wherein said thread is looped back to a beginning stage of said multi-thread execution pipeline loop when said thread reaches an end stage of said multi-thread execution pipeline loop and said thread has not finished processing (OS-9 page 5).

15. Referring to claims 6 and 13, Parady in view of OS-9 has taught wherein said context controller subsystem is further configured to sequence said thread through said miss fulfillment FIFO at a rate having a period substantially equivalent to said pipeline latency (OS-9 page 5). In regards to OS-9, the wait queue only holds processes that are blocked and waiting for some event to complete, e.g. waiting for an I/O access to complete. This I/O access latency substantially equivalent to the pipeline latency.

16. Referring to claims 7 and 14, Parady in view of OS-9 has taught wherein said device request is a request to access external memory due to a cache miss status (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3).

2. Claims 15-17 and 20-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady, U.S. Patent Number 5,933,627 (herein referred to as Parady) in view of Wilford et al., U.S. Patent Number 5,509,006 (herein referred to as Wilford) and in further view of Microware™'s "OS-9: Real-Time Operating System – The Complete Software Solution for Your Embedded Application" ©1999 (herein referred to as OS-9).

3. Referring to claim 15, Parady has taught a fast pattern processor that receives and processes protocol data units (PDUs), comprising:

a. A dynamic random access memory (DRAM) that contains instructions (Parady column 5, lines 19-22; Figure 5; and Figure 6). In regards to Parady, DRAM in a specific type of RAM and Parady shows that RAM is used in his system. Please see Rosenberg's Computers, Information Processing & Telecommunications Second Edition for more information of RAM and DRAM.

b. A memory cache that caches certain of said instructions from said DRAM (Parady column 5, lines 19-22; Figure 5; and Figure 6); and

c. An engine that employs said DRAM and said memory cache to obtain ones of said instructions (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), including:

i. A multi-thread execution pipeline loop having a pipeline latency (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), and

ii. A context switching system for said multi-thread execution pipeline loop (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), having:

(1) A context switch requesting subsystem that: detects a device request from a thread executing within said multi-thread execution pipeline loop for access to a device having a fulfillment latency exceeding said pipeline latency (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), and

(2) Generates a context switch request for said thread (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3), and

iii. A context controller subsystem that receives said context switch request and prevents said thread from executing until said device request is fulfilled (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3).

4. Parady has not taught a tree engine that parses data within said PDUs. Wilford has taught a tree engine that parses data within said PDUs (Wilford column 1, lines 34-42; column 1, line 65 to column 2, line 19; column 14, lines 14-35; and Figure 5B). A person of ordinary skill in the art at the time the invention was made, and as taught in Wilford, would have recognized that a tree engine that parses data within said PDUs identifies which protocol the data belongs to in order to send the data to the correct destination (Wilford column 1, lines 34-42), thereby ensuring correct data execution. Therefore, it would have been obvious to a person of ordinary skill in the art to incorporate the tree engine of Wilford in the device of Parady to ensure correct data execution.

17. In addition, Parady has not explicitly taught

a. A miss fulfillment first-in-first-out buffer (FIFO); and

b. Based thereon, stores said thread in said miss fulfillment FIFO for sequencing there through until said device request is fulfilled.

18. However, Parady has taught using a round robin scheduling method (Parady column 4, lines 9-11). OS-9 has taught round robin scheduling (OS-9 page 5) with a

a. A miss fulfillment first-in-first-out buffer (FIFO) (OS-9 page 5); and

b. Based thereon, store said thread in said miss fulfillment FIFO for sequencing there through to prevent said thread from executing until said device request is fulfilled (OS-9 page 5).

19. In regards to OS-9, the event queue stores processes that are blocked, waiting for some event to complete, like an I/O access. This is similar to a miss fulfillment FIFO, which, by its name and the language in the claim, stores processes that wait for some other event, like an I/O

access, to complete. A person of ordinary skill in the art at the time the invention was made would have recognized that a round-robin scheduling method is a fairly simple method that guarantees that all processes will be executed while ensuring that priority between processes is maintained, thereby preventing starvation of a process while giving consideration to more important processes. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the round robin method of OS-9 in the device to ensure all processes are executed while maintaining priority and simplicity.

20. Referring to claim 16, Parady in view of Wilford and in further view of OS-9 has taught wherein said context controller subsystem further allows a new thread to enter said multi-thread execution pipeline loop after storing said thread in said FIFO (OS-9 page 5). In regards to OS-9, after waiting for the event, e.g. I/O access, to complete, the process is moved to the active queue, which holds processes active and ready to be executed (OS-9 page 5). From the active queue, the highest priority process runs each tick (OS-9 page 5).

21. Referring to claim 17, Parady in view of Wilford and in further view of OS-9 has taught wherein said context controller subsystem further allows other threads within said multi-thread execution pipeline loop to continue to execute while said thread is waiting for said device request to be fulfilled (Parady column 2, lines 28-34; column 3, line 57 to column 4, line 18; column 4, line 42-62; and Figure 3).

22. Referring to claim 18, Parady in view of Wilford and in further view of OS-9 has taught wherein said context controller subsystem is further configured to:

a. Store said thread in said miss fulfillment FIFO upon reaching an end position of said multi-thread execution pipeline loop (OS-9 page 5),

b. Sequence said thread through said miss fulfillment FIFO (OS-9 page 5), and

c. Reinsert said thread into said multi-thread execution pipeline loop at a beginning position (OS-9 page 5).

23. Referring to claim 19, Parady in view of Wilford and in further view of OS-9 has taught wherein said thread is looped back to a beginning stage of said multi-thread execution pipeline loop when said thread reaches an end stage of said multi-thread execution pipeline loop and said thread has not finished processing (OS-9 page 5).

24. Referring to claim 20, Parady in view of Wilford and in further view of OS-9 has taught wherein said context controller subsystem further sequences said thread through said miss fulfillment FIFO at a rate having a period substantially equivalent to said pipeline latency (OS-9 page 5). In regards to OS-9, the wait queue only holds processes that are blocked and waiting for some event to complete, e.g. waiting for an I/O access to complete. This I/O access latency substantially equivalent to the pipeline latency.

5. Referring to claim 21, Parady in view of Wilford and in further view of OS-9 has taught wherein said device request is said DRAM and said device request is a request to access said DRAM due to a cache miss status from said memory cache (Parady Abstract; column 1, lines 29-35 and 46-57; column 2, lines 18-34; column 3, line 57 to column 4, line 18; column 4, lines 42-62; Figure 3).

Response to Arguments

25. The Examiner withdraws objection to claims 5, 12, and 19 in favor of the amended claims.

26. Applicant's arguments filed 06 January 2006 have been fully considered but they are not persuasive. Applicant argues in essence on pages 7-10

> The queues in OS-9, however, are not miss fulfillment FIFO (sic) as recited in independent claims 1 and 8...Unlike the first-in, first-out buffer, processes that enter the queues of OS-9 do not sequence there through but instead, remain in the queue until a condition is met, **regardless** of when the process entered the queue...

27. This has not been found persuasive. Merely adding the language "for sequencing there through" does not cure the deficiencies in the claim. The arguments seem to insinuate that the threads travel through the entire miss fulfillment FIFO, time wise, prior to exiting the FIFO. It merely means that the thread is in the FIFO for some period prior to exiting it, so the fact that "for sequencing there through" was added to the claims does not mean that the claims now recite a clarified limitation that the thread travels through the entire FIFO buffer prior to exiting the FIFO. Also, as stated in the Office Action, the name of the element as well as the description of the element in the OS-9 White Paper is a FIFO type storage method. As shown in the definition from the Free On-line Dictionary of Computing attached, a queue is a first-in first-out data structure, e.g. a FIFO buffer. OS-9 explicitly states "a queue of processes that are currently blocked, waiting for some event value." The event value includes whether or not the thread is the first thread in the queue, and the event value can be this signal alone or in combination with another signal, such as a cache miss being resolved.

Conclusion

28. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

29. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

30. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

31. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

32. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
13 March 2006

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100